Michał Wdowski
Grupa G5b

Metody Numeryczne
Sprawozdanie z Projketu 2

1 Treść zadania

17. Metoda parabol wyznaczania zer wielomianu

$$w_n(x) = a_0U_0(x) + a_1U_1(x) + \cdots + a_nU_n(x),$$

gdzie $U_0, U_1, \ldots, U_n$ są wielomianami Czebyszewa II-go rodzaju.

Uwaga. Nie należy sprowadzać wielomianu w_n do postaci naturalnej! Do obliczania wartości wielomianu w_n oraz jego pochodnych należy wykorzystać związek rekurencyjny spełniany przez wielomiany Czebyszewa.

2 Metoda rozwiązania

Metoda parabol wyznaczania zer wielomianu wygląda w następujący sposób:

- Dane wejściowe:
  - Funkcja rzeczywista f określona i ciągła na przedziale $[a, b]$ $(a < b)$, o której wiemy, że $f(a)$ i $f(b)$ mają różne znaki, zatem z twierdzenia Darboux f ma w przedziale $[a, b$ miejsce zerowe,
  - Liczby rzeczywiste x_0, x_1, x_2 z przedziału $[a, b]$ - początkowe przybliżenia miejsca zerowego,
  - Liczba rzeczywista ε - względnie mała - dopuszczalny błąd wyniku;
- Dane wyjściowe:
  - Liczba rzeczywista x - miejsce zerowe funckji f z dokładnością do ε,
  - Liczba naturalna r - liczba iteracji, która była potrzebna do uzyskania wyniku;
- Sposób działania:
  - Dopóki $|x_2 - x_1| > \varepsilon$ i $|f(x_1)| > \varepsilon$:
    * Znajdujemy parabolę g przechodzącą przez x_0, x_1, x_2,
    * Znajdujemy miejsca zerowe paraboli g i jako x_3 zapisujemy to bliższe x_2,
    * Przypisujemy $x_0 := x_1$, $x_1 := x_2$, $x_2 := x_3$.

Rekurencyjny wzór dla $k = 3, 4, 5, \ldots$ prezentuje się następująco ([1]):

$$x_k = x_{k-1} - \frac{2f(x_{x-1})}{w \pm \sqrt{w^2 - 4f(x_{k-1})f[x_{k-1}, x_{k-2}, x_{k-3}]}},$$

gdzie $w = f[x_{k-1}, x_{k-2}] + f[x_{k-1}, x_{k-3}]] - f[x_{k-2}, x_{k-3}]]$, a zapis $f[x_0, \ldots, x_k]$ oznacza różnicę dzieloną, czyli iloraz różnicowy wyższych niż 2 rzędów ([2]).
We wzorze wybieramy plus lub minus tak, aby mianownik w ułamku był jak największy, a w efekcie otrzymana wartość jak najbliższa x_{k-1}.

Wielomiany Czebyszewa II-go rodzaju definiuje się rekurencyjnie:

- $U_0(x) = 1$, $U_1(x) = 2x$,
- $U_n(x) = 2xU_{n-1}(x) - U_{n-2}(x)$ dla $n = 3, 4, \ldots$.

Liczenie wartości wielomianu będzie się odbywać dynamicznie, na podstawie poprzednich elementów.

3 Testowanie: Obliczanie wartości wielomianów

Metoda wyznaczania wartości wielomianów Czebyszewa II-go rodzaju została zaimplementowana w pliku `w_cz.m`. Przyjmuje ona na wejściu wektor z współczynnikami kolejnych elementów z bazy wielomianów Czebyszewa II-go rodzaju, a także argumnet, dla którego wartość ma być obliczona. Poprawność implementacji niech poprą następujące testy:

- O wielomianach Czebyszewa II-go rodzaju wiemy, że zachodzi dla nich równość $U_n(1) = n+1$. Oznacza to, że suma $U_0(1)+U_1(1)+\cdots+U_{n-1}(1)$ - reprezentowana w argumencie metody `w_cz.m` przez wektor $[1, 1, \ldots, 1] \in \mathbb{R}^n$ - jest równa sumie $1 + 2 + \cdots + n$. Przykładowo - `w_cz([1, 1, 1, 1, 1], 1)` zwraca wynik 15, co zgadza się z naszymi oczekiwaniami.

- Innym faktem o tych wielomianach jest to, że ich pierwiastki to $x_k = \cos(\frac{k\pi}{n+1})$ dla $k = 1, \ldots, n$. Wywołanie funkcji:
  `w_cz([0, 0, 0, 0, 1], [cos(pi/5), cos(2*pi/5), cos(3*pi/5), cos(4*pi/5)])`
  zwraca wyniki:
  `1.0e-15 * [0, -0.1110, 0.5551, -0.8882]`,
  które są blisko zera z błędem mniejszym niż 10^{-15}.

- Możemy spróbować policzyć wartości pewnego wielomianu z bazy wielomianów Czebyszewa II-go rodzaju. Weźmy na przykład wektor testowych współczynników `[2, 1, 3, 7, 4, 2, 0, 6, 9]`. Po ręcznych obliczeniach można to sprowadzić do wielomanu w postaci naturalnej. Wywołanie dwóch funkcji:

  - `w_cz([2, 1, 3, 7, 4, 2, 0, 6, 9], test_x)`,
  - `polyval([2304, 768, -4032, -1088, 2224, 472, -396, -62, 12], test_x)`

  dla testowej wartości `test_x` równej $[1, 0, -1, 1.3, 0.3, -0.7]$, a potem porównanie ich wyników, zwraca wartości `1.0e-12 * [0, 0, 0, 0.9095, 0, 0.0071]`, które sa blisko zera z błędem mniejszym niż 10^{-12}.

Zachodzenie tych własności sugeruje poprawność sposobu implementacji metody.

4 Testowanie: Metoda parabol znajdowania zer

Metoda znajdowania zer funkcji została zaimplementowana w pliku `metoda_parabol.m`, zgodnie z założeniami opisanymi w sekcji **Metoda rozwiązania**.

Tabela 1: Wyniki dla przybliżeń początkowych 0.1, 0.2, 0.3

Maksymalny błąd	Liczba iteracji	Otrzymane przybliżenie	Wynik dla przybliżenia
1e-05	4	0.120777397653651	-2.1017e-09
1e-06	4	0.120777397653651	-2.1017e-09
1e-07	4	0.120777397653651	-2.1017e-09
1e-08	4	0.120777397653651	-2.1017e-09
1e-09	5	0.120777397636576	-8.8818e-16
1e-10	5	0.120777397636576	-8.8818e-16
1e-11	5	0.120777397636576	-8.8818e-16
1e-12	5	0.120777397636576	-8.8818e-16
1e-13	5	0.120777397636576	-8.8818e-16
1e-14	5	0.120777397636576	-8.8818e-16
1e-15	5	0.120777397636576	-8.8818e-16
1e-16	6	0.120777397636576	2.6645e-15
1e-17	8	NaN	NaN

Metoda była testowana dla `w_cz([2, 1, 3, 7, 4, 2, 0, 6, 9], x)`, na potrzeby testów została stworzona funkcja pomocnicza `test_w.m`. Podane początkowe wartości to $x_0 = 0.1, x_1 = 0.2, x_3 = 0.3$. Wyniki w tabeli zależą od maksymalnego błędu.

Liczba iteracji potrzebnych do znalezienia wyniku to mała liczba - maksymalnie 6. Wyniki zbiegają na tyle szybko, że dla mniejszych błędów wystarczający był już otrzymany wcześniej wynik.

Poprawność otrzymanego wyniku można ocenić na podstawie danych w kolumnie "Wynik dla przybliżenia" - jest to po prostu wywołanie `w_cz.m` dla otrzymanego przybliżenia. Wynoszą one niewiele powyżej zera.

W przedostatnim wierszu pojawia się błąd - wynik jest gorszy od poprzednich, mimo że otrzymana wartość jest identyczna. Być może tak naprawdę jest ona inna, a istotą jest problem przedstawienia liczby binarnej w systemie dziesiętnej.

W ostatnim wierszu pojawiają się braki wyników. Dzieje się tak, ponieważ w pewnym momencie w algorytmie metody parabol następuje dzielenie przez różnicę dwóch kolejnych przybliżeń, która zbiega do zera. Wynik staje się tak mały, że przekroczone zostają możliwości MATLABa w przedstawianiu liczb zmiennoprzecinkowych i następuje dzielenie przez zero - wtedy zwracanym wynikiem jest NaN.

Tabela 2: Wyniki dla przybliżeń początkowych 0.4, 0.5, 0.6

Maksymalny błąd	Liczba iteracji	Otrzymane przybliżenie	Wynik dla przybliżenia
1e-05	4	0.514302417604125	5.6246e-11
1e-06	4	0.514302417604125	5.6246e-11
1e-07	4	0.514302417604125	5.6246e-11
1e-08	4	0.514302417604125	5.6246e-11
1e-09	4	0.514302417604125	5.6246e-11
1e-10	4	0.514302417604125	5.6246e-11
1e-11	5	0.514302417603721	-1.3323e-15
1e-12	5	0.514302417603721	-1.3323e-15
1e-13	5	0.514302417603721	-1.3323e-15
1e-14	5	0.514302417603721	-1.3323e-15
1e-15	6	0.514302417603721	-1.3323e-15
1e-16	6	0.514302417603721	-1.3323e-15
1e-17	6	0.514302417603721	-1.3323e-15

W tym wypadku zastosowano początkowe przybliżenia $x_0 = 0.4, x_1 = 0.5, x_3 = 0.6$. Liczba potrzebnych iteracji znowu jest niewielka, ale w tym wypadku metoda okazała się być bardziej stabilna dla bardziej dokładnych przybliżeń.

Ważne dla działania jest odpowiednie dobranie wartości początkowych, ponieważ funkcja może zwracać błędy. Wszystkie początkowe przybliżenia powinny być różne, ale bliskie sobie. Jeśli będą one odpowiednio dobrane, to funkcja potrafi znajdować miejsca zerowe z dobrą dokładnością.

5 Bibliografia

1. https://en.wikipedia.org/wiki/Muller%27s_method
2. https://en.wikipedia.org/wiki/Divided_differences